

General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Memorandum 33-756

*Standard Classifications of
Software Documentation*

R. C. Tausworthe

(NASA-CR-145932) STANDARD CLASSIFICATION OF
SOFTWARE DOCUMENTATION (Jet Propulsion Lab.)
9 p HC \$3.50 CSCL 09B

N76-17854

Unclas

G3/61 13612



JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

January 15, 1976

Preface

The work described in this report was performed by the Telecommunications Division of the Jet Propulsion Laboratory.

Contents

I. Introduction	1
II. Documentation Levels	1
III. Classes and Criteria for Detail in Software Specifications	2
A. Class A Specifications	3
B. Class B Specifications	3
C. Class C Specifications	4
D. Class D Specifications	4
IV. Categories and Criteria for Format Quality in Software Specifications	5
A. Format Category 1: Formal Publication Quality	5
B. Format Category 2: External Report Quality	5
C. Format Category 3: Internal Report Quality	6
D. Format Category 4: Minimal Report Quality	6
V. Conclusions	7
References	7

Figures

1. Documentation-level lattice	2
2. Making a flowchart complete enough for Class A specification of control logic correctness assessment at current level, and for immediate coding	2
3. Example of a Class B detail flowchart	3
4. Example of a Class C detail flowchart, corresponding to the subprogram in Fig. 2	4
5. The FILESORT subprogram represented in Class D detail	5

THIS PAGE BLANK NOT FILMED

Abstract

This report describes general conceptual requirements for standard levels of documentation and for application of these requirements to intended usages. These standards encourage the policy to produce only those forms of documentation that are needed and adequate for the purpose.

Documentation standards are defined with respect to detail and format quality. Classes A through D range, in order, from the most definitive down to the least definitive, and categories 1 through 4 range, in order, from high-quality typeset down to handwritten material. Criteria for each of the classes and categories, as well as suggested selection guidelines for each are given.

Standard Classifications of Software Documentation

I. Introduction

In large, long-life-cycle programming systems, documentation must be provided in considerable detail; however, for small, single-purpose or "one-shot" jobs, hardly any detail may be needed at all. In important or widely used applications, documentation may be typeset, with professionally drafted artwork; in small or exploratory programs, hand-written text and hand-drawn sketches may be sufficient.

Requirements relative to classification of detail, and characterization of the documentation medium normally should be settled prior to the initiation of work. Moreover, such requirements are more easily stated if there are well defined, standard levels of documentation from which to choose. The specified level then becomes the basis for planning, directing, and controlling the documentation effort.

This article describes general conceptual requirements for standard levels of documentation, and for application of these requirements to intended usages. These standards encourage the policy to produce only those forms of documentation that are needed and adequate for the purpose.

II. Documentation Levels

Documentation can be graded into levels, based on information content (extent of detail) and its quality of format and medium. The *level of documentation* for a specific application can therefore be defined in terms of "classes" of detail and "categories" of format quality. This grading can be fitted to the needs of readers and to the available development resources.

Section III below defines four classes of detail, ranging from "Class A" (most definitive) to "Class D" (least definitive). The requirements for each class are successively relaxed for each lower class from A to D. Section IV then defines format-quality categories, ranging from "Format 1" (highest) to "Format 4" (lowest). Selection guidelines for determining appropriate classes and categories also appear in Sections III and IV.

The spectrum of documentation levels, then, defines a lattice structure (Fig. 1) or a set of partial ordering relationships according to detail class and format category. For example, a document graded A1 is both very detailed and beautifully published; a document graded D4 contains little detail and may be hand-written.

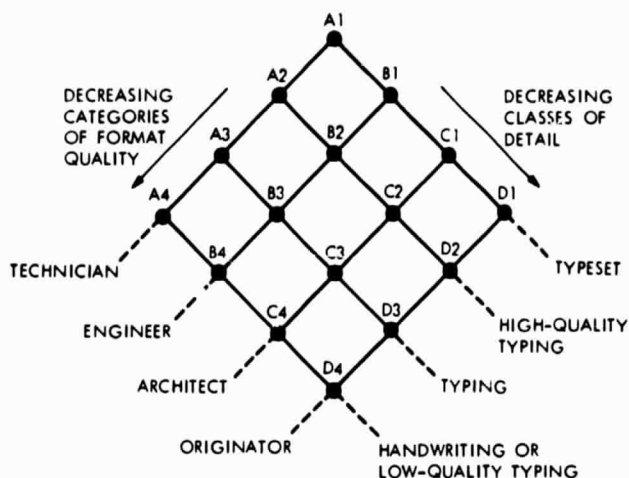


Fig. 1. Documentation-level lattice (Class of detail is graded A,B,C, and D; format quality as 1,2,3, and 4. The expertise of the intended readers for each detail class is indicated for typical specification documents.)

Costs to produce documentation rise as the level of detail increases and also as the format quality increases, and such costs can be estimated with fair accuracy. The costs of *not* having documentation of a given type are not quite so easy to pin down, and depend on the intended life cycle and on down-stream utility factors which may or may not happen to occur. Nevertheless, the agency which sets documentation requirements must carefully estimate and weigh such factors if there is to be a cost-effective documentation plan.

It will therefore be assumed, in the few guidelines given here, that the schedule permits documenting to the appropriate level, and that funding is available for document generation and distribution. The guidelines which follow are meant to help in establishing and selecting an appropriate level for documentation relative to other issues. The principal factors, other than cost and schedule, in selecting a level of needed detail are the assumed levels of skill of the intended readers, the sharing potential of the program, its expected lifetime, its complexity, its use frequency, and its generality of application.

To illustrate the concept of a set of standard documentation levels more definitively, this article will treat a typical Software Specification Document (SSD) as a case in point (Ref. 1). Suitable interpretations for categorizing other types of documentation (requirements, operations, etc.) are then not difficult to imagine.

III. Classes and Criteria for Detail in Software Specifications

The classes defined in this section describe criteria for the amount of detail to be provided when writing a software specification. These classes are consistent with NASA computer documentation guidelines (Ref. 2) but are styled in the manner of JPL standard practices for engineering drawings (Refs. 3, 4).

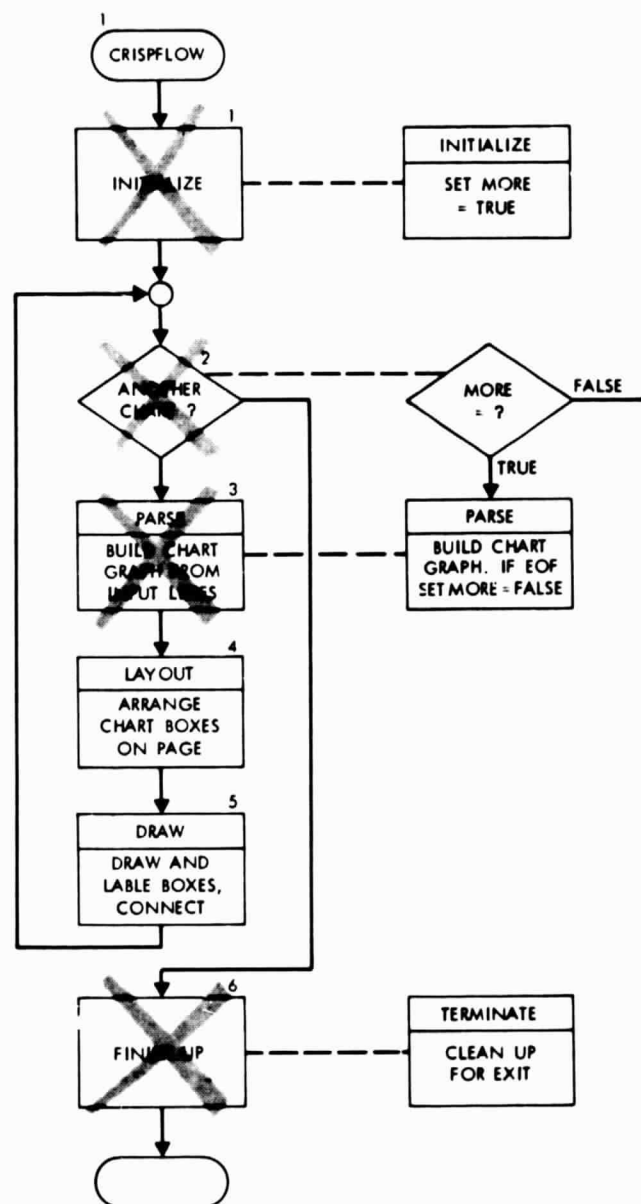


Fig. 2. Making a flowchart complete enough for Class A specification of control logic correctness assessment at current level, and for immediate coding (Crossed-out boxes are to be replaced by those connected by dashed lines. Figure also illustrates Format 1 quality.)

A. Class A Specifications

Class A documentation is the most detailed; it contains specific definitions and detailed descriptions of every significant factor or item within the software specification, so that the program can be understood, implemented, and maintained by any technically qualified personnel (technician or equivalent) without consultation. Every description, function, and operation is specified in that level of detail which permits a correctness assessment on a unit basis and coding without functional ambiguity; no factor of an item contained in Class A documentation is left to the discretion of the implementor. All operations within "unstriped" flowchart symbols (not to be detailed elsewhere in the document; Ref. 5) or similar specifications are to be covered by appropriate references to published works, external standards or internal conventions, or else are at the programming language level (see Fig. 2).

Since the SSD is meant also to be used as a maintenance document, a Class A specification describes all the pertinent aspects of the system, its operational environment, its interfaces, testing, and external data bases, as well as the detailed program functional specification. The level of detail permits an experienced maintenance programmer to correct errors or implement authorized changes without excessive expenditure of time to understand the program (once trained), and without the need for consulting with the program developers.

This class of highest detail is appropriate whenever the intended readers/users are to be relatively unskilled (technician level) personnel and need such detail to perform effectively, or where there is a need to ensure that certain particulars be interpreted in a specific way. In some cases, such as maintenance documentation, there may be a cost advantage in lessening requirements for detail by employing higher levels of skilled personnel. However, it may not always be possible to select or predict the community of readers or users, and in such cases, Class A detail may be appropriate.

This class is also appropriate for SSDs whenever a detailed audit of the program is required or when the document is to be used as a contractual instrument with minimal contract coordination and review. Other indications for selecting this class are (1) a critical application, perhaps involving personal physical risk, (2) good sharing potential, (3) high frequency of use, (4) highly intricate concepts to be communicated, and (5) long or continuing program life cycle.

B. Class B Specifications

Class B documentation requirements are the same as for Class A, except that the requirements for item definition detail are somewhat relaxed. Class B documentation, however, is to be suitable for conversion to Class A quality by the addition of further detail, without extensive effort on the part of the supplier of that detail.

Class B specifications define every factor of the software item being described to the extent that qualified personnel (engineer or equivalent) using documented techniques and approved programming practices can satisfactorily produce that item entirely from information supplied. Some specifications for coding functions, operations, data structures, etc., may be left to the discretion of the implementors if these will satisfy program requirements with respect to performance and quality without unreasonable risk (see Fig. 3 for an example). Assessment of control logic correctness must be possible on an individual unit basis.

The level of detail required for program maintenance also drops to that amount needed by qualified maintenance personnel to correct errors or to implement

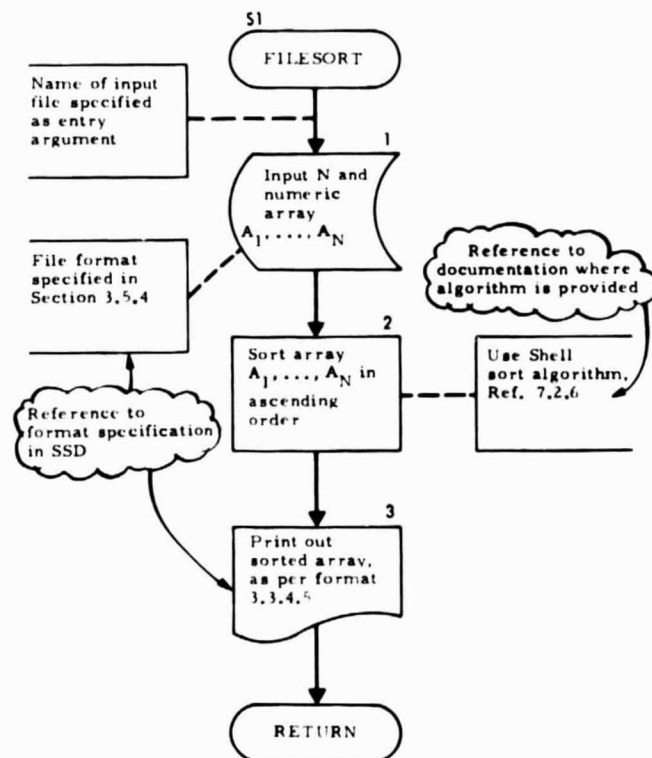


Fig. 3. Example of a Class B detail flowchart: (Text within clouds is not part of the flowchart but an explanation of flowchart conventions. Figure also demonstrates Format 2 quality.)

changes, either after a minimal consultation with the program developers or after some reasonable review and study to discover how a specific part of the program works.

Class B detail applies whenever readers are assumed to have more skill (engineer level) than that deemed appropriate for Class A, or are otherwise likely to respond satisfactorily to less detail. The class is also appropriate for specifications whenever the audit requirements are for consistency only, or when the SSD is to be used as a contractual instrument with moderate vendor coordination and review or with limited contractual risk.

This class applies to normal applications programs with normal usage but with limited sharing potential, or whenever there are perhaps no intricate concepts to be communicated but there is expected to be a long or continuing need for the program and its documentation.

C. Class C Specifications

Class C documentation represents an even further relaxation of requirements for detail than does Class B. Class C documentation may require considerable rewriting to supply detail to meet Class A or Class B standards.

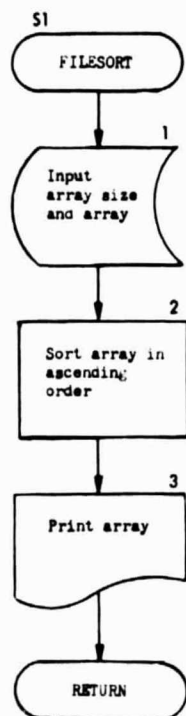


Fig. 4. Example of a Class C detail flowchart, corresponding to the subprogram in Fig. 2 (Format 3 drawing quality is used for illustration.)

Class C documentation of design detail need only extend down to that architectural level sufficient for skilled programmers using hierarchic, modular, structured coding practices to produce an acceptable program. Callouts for standard algorithms, operations, etc., may be used in Class C specifications, leaving the specific methods to the discretion of the programmer, subject to approval by the designers, provided there is minimum increase in risk in not satisfying program requirements with respect to performance and quality and perhaps at a moderate increase in debugging time or exploratory coding (see Fig. 4). However, control logic correctness must still be determinable on an individual module basis.

Class C documentation also further reduces the requirements for detail supplied to maintenance programming. The use of Class C documentation may require more extensive consultation with program developers, or a broader analysis or major reworking of certain parts of the program by the maintenance personnel to correct errors or to make modifications. The minimum documentation required is that necessary to set up the program source deck for operation and modification: I/O formats, setup instructions, and the liberal use of comments in the source deck. (These should be independent of items already explained in the SSD.)

The use of Class C detail in an SSD should be limited to cases involving implementation by highly skilled personnel within the cognizance of the project manager. It may not be generally acceptable or satisfactory as a contractual instrument without close coordination and review of contractor performance.

Class C documentation may also be advisable for uncomplicated programs with anticipated low usage, no sharing potential, and short life expectancy. Class C should probably be chosen to document programs at the architectural or feasibility level, when there are no formal requirements for quality assurance (QA) or audit.

D. Class D Specifications

Class D documentation is the minimum acceptable level of detail advisable for any program whose documentation is meant to be retained and perhaps read by others (or by the implementors at a later time). Such documentation should be prepared only in cases where no upgrading of the documentation class is anticipated, as it may be generally unfeasible to upgrade the classification to Class A, B, or C without a complete redocumentation effort. Standards for minimum detail are at the discretion of the preparer's supervisor.

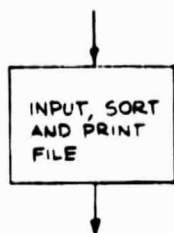


Fig. 5. The FILESORT subprogram represented in Class D detail (also Format 4)

Programs documented as Class D are generally suitable for maintenance only by the original implementor. A conceptual example of a Class D flowchart box appears in Fig. 5.

This class, of least detail, should be limited to summary material, overviews, and other reports of minimal complexity where there is a need to record capability, work done, results, or the function of a program for historical purposes. Class C documentation probably applies to "one-shot" or single-use program SSDs, or to SSDs for programs requiring under 1 manmonth or costing under \$1000. (These are not meant to be equivalent; see Ref. 2.)

Class D documentation should be considered for specification only in those cases where the implementor is well informed of the function and use of the program, as well as system, environment, and other implications. The use of such Class D documentation should usually be restricted to use by the developer only, and may necessitate extensive verbal contact with other readers or extensive time, revision, or rework (due to lack of understanding) by those, other than the developer, who wish to correct errors or to make modifications.

IV. Categories and Criteria for Format Quality in Software Specifications

The categories defined in this section delineate format and publication quality standards for documentation independent of detail classification. Combinations of class and format specifications are comparable with NASA documentation guidelines (Ref. 2) but are more flexible, as the criteria for selection are herein made separate, independent issues.

A. Format Category 1: Formal Publication Quality

Format 1 generally applies to the documentation of programs which are of sufficient general interest, wide

usage, or organizational image value to be announced and distributed in the highest publication quality available. Such documentation should be prepared in a formal, rigorous manner, with in-depth technical review, meticulous proofreading, full editing by professional documentation personnel, and organizational approval for release and distribution.

In most cases, the text of such documents will be *typeset* and permanently bound, or of comparable quality; all artwork will normally be of professional drafting quality (Refs. 3,4) equivalent to inked drawings with high-quality lettering, suitable for "textbook" illustrations. (Figure 2 is an example of such artwork.) Alterations are distributed as errata if minor, and as revisions by reprinting and reissue if major.

Format 1, or formal publication quality, generally applies to cases where the document will find high usage, perhaps external to the organization, where the organizational image plays a role, where there are professional documentation personnel services available for formatting, editing, composing, etc., where there is a general and widespread interest in the program, and where the program is *stable*.

Format 1 documentation probably applies most often to user manuals, announcements and summaries, and perhaps user-group bulletins. It would rarely be used for publication of an SSD, and probably should never be used for a Software Requirements Document (SRD) or Software Design Document (SDD) (Ref. 1).

B. Format Category 2: External Report Quality

Format 2 requirements are the same as those for Format 1, except that the requirements for editing and typography are somewhat relaxed. Format 2 generally applies to the documentation of programs which are expected to be widely used within an organization but may have some readers outside as well. Such documentation should be prepared with adequate technical review, good proofreading, editing sufficient to assure format consistency and clarity of expression, and organizational approval for external release and distribution.

The text of Format 2 documentation should be of *high typewritten quality* and suitable for photo-offset printing, such as that obtained using a 10-point IBM Executive Modern typewriter font. Illustrations and artwork should be drawn to professional ink-line drafting standards (Refs. 3,4), perhaps with typed-in lettering (see Fig. 3). Otherwise, the style is exactly that specified for Format 1. Alterations to the documentation are distributed as change

pages, unless the document is permanently bound, in which case, changes are handled as in Format 1. In any case, the document is normally enclosed in a protective cover.

External Report Quality (Format 2) is the normal level of publication quality for documentation which is expected to find high usage but where professional documentation personnel aid is limited, where the organizational image is less sensitive, where there is less general interest in the program, and where the program documentation is perhaps slightly less stable than one documented as Format 1.

Format 2 documentation probably applies to most of the user manuals, announcements and summaries, and user-group bulletins printed for use within an organization or externally on an interim basis. Use of Format 2 for an SSD will probably be feasible only for highly stable programs whose artwork can be computer-drawn and whose text is contained in, and reproduced by, a computer.

C. Format Category 3: Internal Report Quality

Format 3 is a further reduction in report quality from that required for either of Formats 1 or 2. Format 3 documentation generally applies to special-purpose or in-house programs which, after careful consideration of the possible interests of others, appear to have insufficient usage, sharing potential, or life expectancy to warrant a higher-quality format. Such documentation should be prepared with project and QA review, however. Internal release and distribution are at the discretion of the project manager.

The text of Format 2 documentation should be typewritten, although there need not be any requirement placed on the typewriter font. Artwork may be hand-drawn (in pencil, if reproducible) using standard templates and typed-in lettering (see Fig. 4, for example). There is no relaxation on the style of narratives or illustrations, however, from Formats 1 and 2. Any reproduction medium and binding suitable to the limited distribution are permissible. Covers are arbitrary.

Alterations to Format 3 documentation are handled by distributing change pages.

The normal working level of documentation within an organization is Format 3. It is generally used when the numbers of users of the documentation are limited but where there is a need for continued use or a permanent record of the recorded items. The documentation is

usually prepared within the implementing organization, without the aid of professional documentation personnel, and where there are limited funds or facilities for drafting and other artwork. The format, being less restricted, can also accommodate a somewhat less stable programming environment.

Format 3 documentation is applicable to the working-level SSD, SDD, and SRD, as well as low-use documents designed for program maintenance and operations.

D. Format Category 4: Minimal Report Quality

Format 4 is the lowest quality of documentation, and the least restricted. Format 4 probably applies most often to single-use programs, or "one-shot" jobs, of minimal complexity but for which there exists a requirement to report or record the type of work being produced or the results of a given effort for historical purposes. Review, QA, and distribution are at the discretion of the developer(s).

Format 4 documentation may be freehand (Fig. 5) or may deviate from style requirements and standard practices to whatever extent practical, at the discretion of the preparer's supervisor. Format 4 documentation need meet only the minimum requirements necessary for storage and retrieval in the Software Development Library, if such facility is used. (It is usually desirable to keep on file for some period of time the documentation which results from program development, such as a program abstract, the project notebook, a compiled source listing, test cases, run examples, etc.)

Alteration methods are at the discretion of the preparer or his supervisor.

Format 4 is suitable for exploratory or look-ahead efforts, where there is little distribution potential, or for programs requiring less than 1 manmonth of effort or costing under \$1000. (These are not meant to be equivalent; see Ref. 2.) Because of the informality of the quality restrictions, however, Format 4 documentation can probably accommodate, better than other formats, the dynamic needs for documenting any unstable elements in the system.

In most projects leading to operational programs, Format 4 documentation is generally not suitable for any forms of program documentation, except perhaps project bulletins, status reports, memos, etc.

V. Conclusions

Adequate documentation of computer programs is clearly an essential element of efficient and economical use of computer systems. Good documentation prevents waste and unnecessary costs in many ways—by making program modifications feasible, by making redesigns easier, by making internal controls work better, by facilitating the work of auditors, and in a host of other ways, all equivalent to making programs usable by others.

Lack of needed documentation and low-quality documentation are problems in human engineering which in large part can be remedied by setting good standards and then seeing to it that work necessary for documentation is performed according to those standards. This report contributes to the solution of such problems by outlining standard levels of documentation by content and format, and giving typical criteria for choosing among these documentation levels.

References

1. *Software Implementation Guidelines and Practices*, DSN Standard Practice 810-13, Jet Propulsion Laboratory, Pasadena, Calif., June 1975 (JPL internal document).
2. *Computer Program Documentation Guideline*, NHB 2411.1, National Aeronautics and Space Administration, July 1971.
3. *JPL Drafting Manual*, Document JPL-STD00001A, Jet Propulsion Laboratory, Pasadena, Calif., September 15, 1969 (JPL internal document).
4. *Military Standard Engineering Drawing Practices*, MIL-STD-100A, Department of Defense, Washington, D.C., October 1, 1967.
5. *American National Standard Flowchart Symbols and Their Usage in Information Processing*, ANSI-X3.5-1970, American National Standard Institute, Inc., September 1, 1970.